

Art Unit: 2122

IV. Remarks

Claim 10 was rejected under 35 U.S.C. §112, second paragraph, as being indefinite. Claims 1-11 were rejected under 35 U.S.C. §102(a) as being anticipated by Bohrer. Reconsideration is respectfully requested.

Claim 10 was rejected as being indefinite. Claim 10 has been amended to correct the language noted in the Office Action. Thus, the rejection of claim 10 has been overcome and should be withdrawn.

Claims 1-11 were rejected as being anticipated by Bohrer. Applicant respectfully asserts that the claims patentably distinguish over the cited art. First, Bohrer is not prior art. Second, Bohrer does not disclose, teach or suggest the claimed invention. Accordingly, the rejection of claims 1-22 as being anticipated by Bohrer should be withdrawn.

The Office Action relies upon Bohrer as prior art under 35 U.S.C. §102(a). Bohrer issued on August 22, 2000 based upon an application filed on March 11, 1998. The present application was filed on November 2, 2001 and claims priority to a Provisional Patent Application filed on August 3, 2001. Thus, the priority date is less than one year before the issue date of Bohrer, and Bohrer cannot be prior art under 35 U.S.C. §102(b). Additionally, Bohrer does not predate the invention date of the present invention, as required for prior art under 35 U.S.C. §102(a). Mr. Shiu conceived of the key aspects of the invention in 1996. On June 10, 1996, he formed a company to create a development system based upon his conception. The key aspects of the invention were illustrated in documents as early as February 1997. The development effort continued through the end of 2000, when the system was first used for commercial development of software applications. The system has continued to be modified and improved since that time. Based upon the dates of conception and reduction to practice, and the ongoing development of the system, the invention date of the present invention predates both the issuance and filing dates of Bohrer. Accordingly, Bohrer is not prior art and the rejection of claims 1-11 under 35 U.S.C. §102(a) as being anticipated by Bohrer should be withdrawn.

Furthermore, even if Bohrer were prior art, it does not anticipate the present claimed invention. While both Bohrer and the present invention relate to systems for developing

**Art Unit: 2122**

software, they do it using completely different processes. Bohrer relates to a method for developing software using standard object oriented technology and frameworks. Bohrer includes processes which allow objects to acquire and lose ability and function and to modify responsibilities on an object dynamically. Software is developed using a framework which includes "Life Cycles" to define transitions through which an object can proceed as processed by an application. The framework also provides significant portions of the application which can be extended to specific implementations upon creation of an application.

Although the present application refers to "objects", it does not use the term in the manner of object oriented technology. The present invention provides a development system based upon data model or object model. An object model (or data model, which are used synonymously) as set forth in the present application describes data structures representing application data as basic data patterns or types. Specifically, in the present invention, all data in an application is represented as data models. A data model is a set of basic data types, which may include primitives, classes and arrays. A data model does not correspond to an "object" in object oriented technology. An object has attributes and values. Objects may also have functions. Furthermore, objects have inheritance. Finally, objects include specific instances of object classes. As discussed in Bohrer, object oriented technology includes objects as "self-contained piece[s] of software having data and procedures." Column 2, lines 15-16. The data model or object model of the present invention does not include any of these features. Thus, although similar terms are used, they have different meanings.

Similarly, the "service objects" of the present invention are unrelated to object oriented technology. In the present invention, service objects define functions to be performed on the object models. The service object defines the functions for the basic object types within a object model. During runtime, the service object recognizes the basic object types with a data model and performs the functions corresponding to the basic object types in the data model. Thus, the service objects, unlike traditional programming and object oriented technology, are independent of the application data.

Finally, the present invention includes flow processes which represent an order for operation of the service objects. In order to develop software using the present invention, the

Art Unit: 2122

developer creates object models for the application data and selects the service objects necessary to perform the desired functions. The flow process selects the next service object to act on the data. The service object recognizes the basic data types in the object model and performs the programmed functions for data of those types.

Claim 1 has been amended to clarify the terminology of the present application. Claim 1, as originally written and as amended, patentably distinguishes over Bohrer. Claim 1 recites a method for creating application software and includes three steps. The first step is storing a set of object models representing the application data. Each object mode corresponds to a set of basic object types. As discussed above, Bohrer and all other object oriented technology development software do not store a set of object models representing application data within the meaning of the present invention.

The second step is selecting a subset of service objects from a stored set of service objects. The service objects perform a function with respect to data of the basic object types. Bohrer does not disclose, teach or suggest selection of service objects within the meaning of the present claimed invention. The Office Action references the Common Business Object layer 102 and Core Business Process layer 103 in Bohrer as corresponding to the service objects of the present invention. However, these layers of Bohrer have a different design and operation from the service objects of the present invention. The Common Business Objects are objects within the meaning of object oriented technology in that they include data and processes. In particular, they perform common functions within a business application, such as date and time, currency, address, units of measure and calendar. (column 6, lines 55-56). Thus, the functions are specifically tied to the format of the data. The Core Business Process layer “contains the basic functions which all of the [business] application programs ... require.” (column 7, lines 6-8). The Core Business Process layer creates frameworks linking the Common Business Objects “to a large number of objects specific to the type of framework being built.” (column 7, lines 10-11). Thus, Bohrer uses several levels of objects, each including data and processes, in various layers. Further, Bohrer requires a large number of objects specific to the application within the framework.

Art Unit: 2122

On the other hand, the service objects of the present invention provide functionality based upon the basic object types. In the present invention, as recited in claim 1, the service objects are detached from the data and the format of the data. The service objects operate on all data of a basic object type. Thus, each service object can be used with any data model. Bohrer does not disclose, teach or suggest selecting a subset of service objects as recited in claim 1.

Finally, claim 1 recites defining a flow process representing an order of operation of the subset of service objects. Since Bohrer does not use service objects, it cannot disclose defining the flow process for their order of operation. The Office Action references the Life Cycle and extendable methods in Bohrer as illustrating this feature of claim 1. However, the Life Cycle and extendable methods in Bohrer relate to changes in the functionality of certain objects in the object oriented technology. The flow process, as recited in claim 1, does not relate to changed functionality. Claim 1 recites that the flow process defines an order of operation of the selected subset of service objects. Bohrer does not disclose, teach or suggest defining an order of operation of the subset of service objects.

Since Bohrer does not disclose, teach or suggest any of the steps of method claim 1, the rejection of claim 1 as being anticipated by Bohrer is improper and should be withdrawn. Claims 2-7 depend from claim 1 and are allowable for at least the same reasons.

Claim 8 recites a system for creating application software. The system includes means for storing a set of objection models, a stored set of service objects, means for selecting a subset of service objects, and means for defining a flow process. As discussed above, Bohrer does not disclose, teach or suggest, object models, service objects or flow processes within the meaning of the present claimed invention. Accordingly, claim 8 patentably distinguishes over the cited art and is in condition for allowance.

Claims 9 and 10 depend from claim 8 and are allowable for at least the same reasons. Furthermore, claim 9 recites that the means for storing the object models includes: means for receiving an application model, means for classifying each data element in the application model as a object model, and means for storing each object model. Bohrer does not disclose, teach or suggest such a system. The Bohrer framework uses Common Business Objects and a large number of application specific objects to create an application. There is not an application

Art Unit: 2122

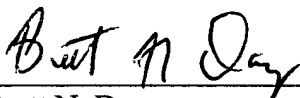
model, and nothing in Bohrer receives an application model. Nothing in Bohrer classifies each data element in an application model as an object model, and there are not object models.

Therefore, claim 9 patentably distinguishes over the cited art.

Claim 11 recites a system for executing an application program. The system comprises a stored set of object models, an stored set of service objects, means for determining the basic object types of an object model upon execution of the service objects, and means for executing a function of each service object based upon the basic object types. As discussed above, Bohrer does not disclose, teach or suggest, object models or service objects within the meaning of the present invention. Furthermore, Bohrer does not disclose determining the basic object types of a data model and performing a function of a service object based upon the basic object types. The objects in Bohrer include the data and procedures. There are no basic data types in Bohrer, and no mechanism for determining basic data types within an object. The functions of the objects in Bohrer do not depend upon a determination of the basic object types. Accordingly, claim 11 patentably distinguishes over the cited art and is in condition for allowance.

In view of the foregoing amendments and remarks, applicants believe that this application is in condition for allowance. If the examiner has any questions regarding this amendment or the application in general, he is encouraged to telephone the undersigned attorney so that prosecution of this application can be expedited.

Respectfully submitted,



---

Brett N. Dorny  
Registration No. 35,860  
Law Office of Brett N. Dorny  
321 Church Street  
Northborough, Massachusetts 01532  
Telephone (508) 904-3228  
Attorney for Applicant

Date: September 1, 2004

*Correspondence should be addressed to customer number 43,891.*